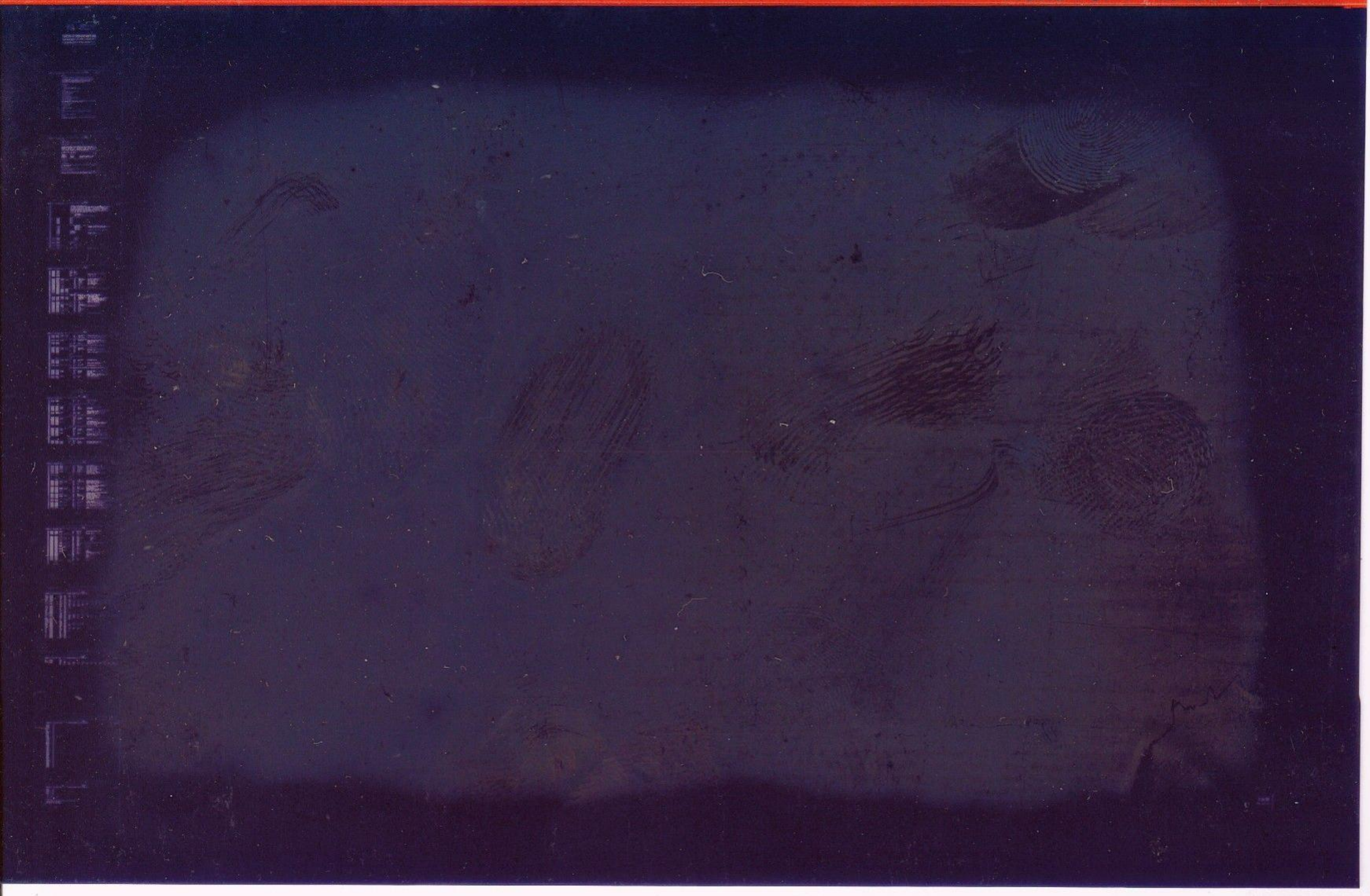
KJ11-A

11/40,45 STACK LIMIT
CCKBFD0

AH-7804D-MC

FICHE 1 OF 1

MAR 1977
digital
MADE IN USA



.REM %

IDENTIFICATION

PRODUCT CODE:	AC-7802D-MC
PRODUCT NAME:	CCKBFDO 11/40,45 STACK LIMIT
PRODUCT DATE:	FEB 1978
MAINTAINER:	PRODUCT ENHANCEMENT

1.0 ABSTRACT

THIS PROGRAM INCREMENTLY TESTS THE STACK LIMIT FUNCTION.

THE PROGRAM USES THE CONTENTS OF LOCATION 176 AS THE VALUE OF THE SWITCHES IF NO HARDWARE SWITCH REGISTER IS FOUND. THE OPERATOR IS RESPONSIBLE FOR LOADING THE DESIRED VALUE BEFORE STARTING THE PROGRAM. LOCATION 174 WILL BE USED AS THE SOFTWARE DISPLAY REGISTER.

2.0 REQUIREMENTS

2.1 EQUIPMENT

BASIC 11/45 SYSTEM OR
11/40 WITH STACK LIMIT OPTION

2.2 STORAGE

THIS PROGRAM USES 0 THRU 17500

2.3 PRELIMINARY PROGRAMS

DOAA THRU DOMA

3.0 LOADING PROCEDURE

LOAD PROGRAM USING ABS LOADER.

4.0 STARTING PROCEDURE

LOAD ADDRESS 200, PRESS START, THE PROGRAM WILL LOOP AND RING BELL AND PRINT '&' ON PASS COMPLETION.

5.0 OPERATING PROCEDURE

5.1 SWITCH SETTINGS

NONE

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

SCOPE IS A "MOVE PC,R1" WHICH STORES THE PC+2 IN R1.

5.2.2 HLT

HLT IS A HALT INSTRUCTION.

6.0 ERRORS

ALL ERRORS WILL CAUSE A HALT. TRAP AND INTERRUPT ERRORS WILL CAUSE A HALT AT VECTOR+2.

6.1 ERROR RECOVERY

PRESS CONTINUE TO PROCEED TO NEXT TEST.

6.2 ERROR LOOPING

TO LOOP ON AN ERROR, PLACE A BRANCH TO THE PREVIOUS SCOPE INSTRUCTION IN PLACE OF THE HALT INSTRUCTION. NOTE THAT IF THE ERROR IS INTERMITTANT THAT THE TEST WILL DROP THRU THE HALT AND PROCEED TO THE NEXT TEST. THEREFORE, TO LOOP THE TEST CONTINUOUSLY REPLACE THE BEQ .+4 INSTRUCTION IMMEDIATELY PRECEEDING THE HALT WITH A BRANCH BACK TO THE PREVIOUS SCOPE.

TO LOOP ON TRAP FAILURES, PATCH IN THE FOLLOWING ROUTINE AT THE ADDRESS OF THE TRAP VECTOR.

```
TRAPVEC:        TRAPVEC+4
TRAPVEC+2:      0
TRAPVEC+4:      012716          ;MOVE SCOPE ADDRESS TO STACK
TRAPVEC+6:      ADDRESS         ;ADDRESS OF PREVIOUS SCOPE
TRAPVEC+10:     000006          ;RETURN TO TEST AT SCOPE
```

RESTORE ALL LOCATIONS BEFORE PROCEEDING TO NEXT TEST.

7.0 RESTRICTIONS

NONE

8.0 MISCELLANEOUS

ON TRAP ERRORS THE STACK POINTER (R6) WILL CONTAIN ADDRESS WHERE THE TRAP OCCURRED.

8.1 EXECUTION TIME

EACH PROGRAM TAKES ABOUT 1 MINUTE.

9.0 PROGRAM DESCRIPTIONS

THIS IS A TEST OF THE STACK LIMIT REGISTER AND INSURES CORRECT OPERATION OF THE RED AND YELLOW ZONE BOUNDARIES. OVERFLOW TRAPS ARE TESTED FOR ALL VALUES OF THE STACK LIMIT REGISTER.

%

```
                                ;CCKBFD0 11/40,45 STACK LIMIT
                                ;THE STACK LIMIT REGISTER ALLOWS THE 'OVERFLOW' BOUNDARIES TO BE CHANGED.
                                ;FOR EXAMPLE IF THE STACK LIMIT REGISTER IS CLEAR THE BOUNDARY IS AT
                                ;400 (YELLOW ZONE) AND 340 (RED ZONE).  IN ALL CASES THE YELLOW ZONE
                                ;BOUNDARY IS AT 400(8) PLUS THE VALUE IN THE STACK LIMIT REGISTER, AND
                                ;THE RED ZONE BEGINS 20(8) WORDS BELOW THE YELLOW ZONE.  THIS TEST
                                ;CHECKS THAT THE STACK LIMIT IS 400 GREATER THAN THE CONTENTS OF THE
                                ;STACK LIMIT REGISTER (CORE PERMITTING), AND CHECKS THE LENGTH OF THE
                                ;YELLOW ZONE AND THE BEGINNING OF THE RED ZONE.

                                ;STARTING PROCEDURE
                                ;       LOAD ADDRESS=200
                                ;       PRESS START
                                ;       BELL WILL RING WHEN TEST IS COMPLETE

                                ;EQUATE STATEMENTS
000000                          R0=%0
000001                          R1=%1
000002                          R2=%2
000003                          R3=%3
000004                          R4=%4
000005                          R5=%5
000006                          SP=%6
000007                          PC=%7

                                ;REGISTER ADDRESSES
177776                                  PSW=177776              ;ADDRESS OF PROCESSOR STATUS WORD
177770                                  UBREAK=177770           ;ADDRESS OF PDP11/45 MICRO BREAK REGISTER
177570                                  DSWR=177570             ;ADDRESS CONSOLE SWITCH REGISTER
177564                                  TPS=177564
177566                                  TPB=177566
177570                                  DDISP=177570            ;ADDRESS OF CONSOLE DISPLAY REGISTER
000000                          HLT=HALT
022626                          POP2=22626
010701                          SCOPE=010701                    ;MOVE PC TO R1
000340                          PRTY7=340                       ;PRIORITY LEVEL 7
000004                          ERRVEC=4                        ;ADDRESS OF ERROR VECTOR

000000                                  .=0


000046                                  .=46
000046  002344                          $ENDAD
000052                                  .=52
000052  000000                          000000

000174                                  .=174
000174  000000                  DISPREG:0
000176  000000                  SWREG:  0

000200                                  .=200
000200  000167  000604                  JMP     START

000500                                  .=500
                                ;TAGS
000500  000000                  ICNT:   0                       ;CONTAINS PASS COUNT
```

```
                                ICNTA:  0
                                SWR:    DSWR
                                DISPLAY:DDISP
                                SPBOT:  760
                                SLR:    177774                  ;ADDRESS OF STACK LIMIT REGISTER
                                SLH:    177775                  ;HIGH (ODD BYTE)
                                TEMP:   0
                                FTITLE: 0                       ;TITLE PRINTED = 1

                        001010                  .=1010
001010  016706  177474          START:  MOV     SPBOT,%6        ;INITIALIZE STACK POINTER
001014  012737  000340  177776          MOV     #PRTY?,@#PSW    ;LOCK OUT INTERRUPTS
001022  023737  000042  000046          CMP     @#42,@#46       ;ARE WE IN ACT11 AUTOMATIC MODE?
001030  001475                          BEQ     STARTB          ;YES, SKIP THE TITLE
001032  005767  177462                  TST     FTITLE          ;TITLE PRINTED YET?
001036  001034                          BNE     STARTA          ;YES, SKIP TITLE
001040  012700  001064                  MOV     #TITLE,R0       ;GET MESSAGE ADDRESS
001044  105710                  1$:     TSTB    (0)             ;END OF MESSAGE?
001046  001430                          BEQ     STARTA          ;YES, GET OVER THE ASCII
001050  105767  176510                  TSTB    TPS
001054  100375                          BPL     .-4
001056  112067  176504                  MOVB    (0)+,TPB        ;PRINT CHARACTER
001062  000770                          BR      1$
001064  005015  041577  045503  TITLE:  .ASCIZ  <15><12><177>?CCKBFD0 11/40,45 STACK LIMIT?<15><12><177>
001072  043102  030104  030440
001100  027461  030064  032054
001106  020065  052123  041501
001114  020113  044514  044515
001122  006524  077412     000
                        001130          .EVEN
001130  012767  000001  177362  STARTA: MOV     #1,FTITLE       ;SET TITLE PRINTED FLAG
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
001136  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
001142  012737  001176  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
001150  012767  177570  177326          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
001156  012767  177570  177322          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
001164  022777  177777  177312          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
001172  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
001174  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
001176  012716  001204          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
001202  000002                          RTI
001204  012767  000176  177272  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
001212  012767  000174  177266          MOV     #DISPREG,DISPLAY
001220  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
001224  005067  177250          STARTB: CLR     ICNT            ;CLEAR PASS COUNT
001230  005767  177246                  TST     ICNTA
001234  001412                          BEQ     BEGIN
001236  022767  177777  001104          CMP     #-1,$ENDAD+4
001244  001006                          BNE     BEGIN
001246  022767  001477  177224          CMP     #1477,ICNT
001254  001002                          BNE     BEGIN
001256  005067  177220                  CLR     ICNTA

001262  016706  177222          BEGIN:  MOV     SPBOT,%6        ;INITIALIZE STACK POINTER
```

```
                                ;CHECK THAT CP CAN TIME OUT TRAP
001312  012737  001330  000004        MOV     #TORET,@#ERRVEC ;LOAD TIMEOUT TRAP VECTOR
001320  005037  173000                CLR     @#173000        ;ADDRESS 173000 ALWAYS TIMES OUT ON
                                                              ;DATIP/DATO BUS CYCLE
001324  000000                        HLT                     ;ERROR! FAILED TO TIME OUT TRAP
001326  000755                        BR      BEGIN           ;LOOP TEST
001330  022626                TORET:  CMP     (6)+,(6)+       ;RESTORE THE STACK

                                ;TEST THAT THE STACK LIMIT REGISTER CAN BE REFERENCED USING DATI,
                                ;DATIP/DATO
001332  010701                TO:     SCOPE
001334  012767  001356  176442        MOV     #TOA,ERRVEC     ;LOAD ERROR VECTOR
001342  005067  176440                CLR     ERRVEC+2
001346  017737  177140  177774        MOV     @SLR,@#177774   ;REFERENCE STACK LIMIT REGISTER
001354  000415                        BR      TOB             ;GO TO NEXT TEST
001356  022626                TOA:    POP2
001360  000000                        HLT                     ;ERROR:  CANNOT REFERENCE STACK LIMIT REG.
001362  000763                        BR      TO              ;LOOP TEST IF ERROR

        001364                  STKL01=.

                                ;USING DATI, DATIP/DATOB

        001410                  .=1410

001410  012767  001426  176366  TOB:    MOV     #TOC,ERRVEC     ;LOAD ERROR VECTOR
001416  117777  177070  177070        MOVB    @SLR,@SLH       ;REFERENCE ODD BYTE
001424  000403                        BR      TOD
001426  022626                TOC:    POP2
001430  000000                        HLT                     ;ERROR! CANNOT REFERENCE STACK LIMIT
001432  000766                        BR      TOB             ;USING BYTE INSTRUCTION.

001434  012767  000006  176342  TOD:    MOV     #6,ERRVEC       ;RESTORE ERROR TRAP VECTOR
001442  000400                        BR      T1              ;GO TO NEXT TEST

                                ;TEST THAT EACH BIT OF THE STACK LIMIT REGISTER BITS CAN BE SET
                                ;AND CLEARED.  THIS TEST ROTATES A BIT THROUGH THE STACK LIMIT REGISTER.

001444  010701                T1:     SCOPE
001446  012702  000400                MOV     #400,R2         ;LOAD TEST VALUE
001452  010277  177034        T1A:    MOV     R2,@SLR         ;LOAD TEST VALUE INTO STACK LIM. REG.
001456  017700  177030                MOV     @SLR,R0         ;GET RESULT
001462  020002                        CMP     R0,R2           ;CHECK RESULT
001464  001401                        BEQ     .+4             ;BRANCH IF RESULT IS CORRECT
001466  000000                        HLT                     ;ERROR! INCORRECT RESULT. R2 HAS CORRECT
                                                              ;RESULT AND R0 HAS INCORRECT RESULT.
001470  005077  177016                CLR     @SLR            ;CLEAR STACK LIM. REG.
001474  017700  177012                MOV     @SLR,R0         ;GET AND CHECK RESULT
001500  001401                        BEQ     .+4
001502  000000                        HLT                     ;ERROR! INCORRECT RESULT
```

```
                                                ASL     R2              ;SHIFT TEST VALUE
                                                BCC     T1A             ;BRANCH IF NOT DONE
                                                BR      T2
                                        ;THIS TEST INCREMENTS THE STACK LIMIT REGISTER
                                        T2:     SCOPE
                                                CLR     TEMP
                                                CLR     @SLR            ;CLEAR STACK LIMIT REGISTER
                                        T2A:    MOV     @SLR,R0         ;GET RESULT
                                                CMP     R0,TEMP         ;CHECK RESULT
                                                BEQ     .+4
                                                HLT                     ;ERROR! STACK LIM. REG. WAS INCORRECT DATA
                                                                        ;TEMP # CORRECT RESULT
                                                INCB    @SLH            ;INCREMENT VALUE IN STACK LIM. REG.
                                                INCB    TEMP+1          ;INCREMENT TEST VALUE
                                                BNE     T2A             ;BRANCH IF ALL VALUES NOT TESTED
                                        ;TEST THAT RESET CLEARS THE STACK LIMIT REGISTER
                                        T3:     SCOPE
                                                MOV     #-1,@SLR        ;PRESET SLR
                                                MOV     @SLR,R0         ;SAVE SLR
                                                CMP     #177400,R0      ;CHECK THAT SLR WAS LOADED
                                                BEQ     .+4
                                                HLT                     ;! SLR FAILED TO LOAD
                                                RESET                   ;RESET CLEARS SLR
                                                MOV     @SLR,R0         ;GET RESULT OF RESET AND CHECK RESULT
                                                BEQ     T4              ;GO TO NEXT TEST IF RESET CLEARED SLR
                                                HLT                     ;ERROR! RESET FAILED TO CLEAR SLR
                                                BR      T3              ;LOOP TEST IF ERROR
                                        ;TEST THAT THE CLEAR INSTRUCTION CLEARS THE STACK LIMIT REGISTER
                                        T4:     SCOPE
                                                MOVB    #-1,@SLH        ;PRESET ODD BYTE
                                                MOV     @SLR,R0         ;GET RESULT
                                                CMP     #177400,R0      ;CHECK RESULT
                                                BEQ     .+4
                                                HLT                     ;ERROR! SLR DID NOT PRESET
                                                CLR     @SLR
                                                MOV     @SLR,R0         ;GET RESULT OF CLEAR & BRANCH IF CLEAR
                                                BEQ     T5              ;GO TO NEXT TEST
                                                HLT                     ;ERROR! CLR INST FAILED TO CLEAR SLR
                                                BR      T4              ;LOOP TEST IF ERROR
                                        ;TEST THAT AN OVERFLOW ERROR OCCURS FOR ALL STACK LIMIT REGISTER VALUES.
                                        ;(PROVIDED CORE IS AVAILABLE).
                                        T5:     SCOPE
                                                MOV     #10,R2          ;INITIALIZE STACK VALUE AND
                                                MOV     #-400,R3        ;STACK LIMIT REGISTER VALUE
                                        T5A:    ADD     #400,R2         ;LOAD NEW STACK VALUE
                                                ADD     #400,R3         ;AND NEW STACK LIM. REG. VALUE
```

```
372  001676  005037  000000                    CLR     @#0             ;CLEAR ADDRESS 0
373  001702  016706  176602                    MOV     SPBOT,SP        ;INITIALIZE THE STACK POINTER
374  001706  012767  002206  176070  T5B:      MOV     #LIMX,ERRVEC    ;LOAD TIME OUT TRAP
375  001714  016204  177776                    MOV     -2(2),R4        ;SAVE STACK LOCATIONS
376  001720  016205  177774                    MOV     -4(2),R5        ;EXIT TEST IF EITHER INST TIMES OUT
377  001724  012767  001750  176052            MOV     #LIMA,ERRVEC    ;LOAD OVERFLOW VECTOR
378  001732  010206                            MOV     R2,SP           ;LOAD STACK POINTER
379  001734  010377  176552                    MOV     R3,@SLR         ;LOAD STACK LIM. REG.
380  001740  016666  177770  177770            MOV     -10(6),-10(6)   ;REFERENCE LIMIT ADDRESS
381  001746  000420                            BR      LIMAA           ;SHOULD NOT HAVE TRAPPED
382  001750  000000                    LIMA:   HLT                     ;ERROR! REFERENCE TO LIMIT ADDRESS
383                                                                    ;CAUSED AN OVERFLOW
384  001752  000416                            BR      LIMAA
385
386          001754                    STKL02=.
387
388
389          002010                    .=2010
390
391  002010  010206                    LIMAA:  MOV     R2,SP           ;REPOINT STACK POINTER
392  002012  012767  002030  175764            MOV     #LIMB,ERRVEC    ;REPOINT OVERFLOW VECTOR
393  002020  016666  177766  177766            MOV     -12(6),-12(6)   ;REFERENCE FIRST 'YELLOW' ADDRESS
394  002026  000000                            HLT                     ;ERROR! SHOULD HAVE TRAPPED
395  002030  005737  000000            LIMB:   TST     @#0             ;WAS IT 'RED' OVERFLOW?
396  002034  001401                            BEQ     .+4
397  002036  000000                            HLT                     ;ERROR! A 'RED' OVERFLOW OCCURRED WHEN
398                                                                    ;A 'YELLOW' ADDRESS WAS REFERENCED.
399  002040  010206                            MOV     R2,SP           ;REPOINT STACK POINTER
400  002042  005037  000000                    CLR     @#0             ;CLEAR ADDRESS 0
401  002046  012767  002064  175730            MOV     #LIMC,ERRVEC    ;REPOINT OVERFLOW VECTOR
402  002054  016666  177730  177730            MOV     -50(6),-50(6)   ;REFERENCE LAST 'YELLOW' ADDRESS
403  002062  000000                            HLT                     ;ERROR! SHOULD HAVE TRAPPED
404  002064  005737  000000            LIMC:   TST     @#0             ;WAS IT 'RED' OVERFLOW
405  002070  001401                            BEQ     .+4
406  002072  000000                            HLT                     ;ERROR! A 'RED' OVERFLOW OCCURRED WHEN
407                                                                    ;THE LAST 'YELLOW' ADDRESS WAS REFERENCED
408  002074  005037  000000                    CLR     @#0             ;CLEAR ADDRESS 0
409  002100  010206                            MOV     R2,SP           ;REPOINT THE STACK POINTER
410  002102  012767  002124  175674            MOV     #LIMD,ERRVEC    ;REPOINT THE OVERFLOW TRAP
411  002110  016267  177726  176400            MOV     -52(2),TEMP     ;GET 'RED' LOCATION
412  002116  005166  177726                    COM     -52(6)          ;REFERENCE 'RED' ADDRESS
413  002122  000000                    LIMCC:  HLT                     ;ERROR! NO OVERFLOW TRAP WHEN 'RED' ADDRESS
414                                                                    ;WAS REFERENCED
415  002124  026267  177726  176364  LIMD:   CMP     -52(2),TEMP     ;WAS INSTRUCTION ABORTED?
416  002132  001401                            BEQ     .+4
417  002134  000000                            HLT                     ;ERROR! COM -52(6) WAS ALLOWED TO CHANGE
418                                                                    ;A 'RED' ADDRESS.
419  002136  016762  176354  177726            MOV     TEMP,-52(2)     ;RESTORE 'RED' LOCATION IN ANY EVENT
420  002144  005706                            TST     %6              ;WAS STACK POINTER CHANGED TO 0?
421  002146  001401                            BEQ     .+4
422  002150  000000                            HLT                     ;'RED' OVERFLOW DID NOT ASSUME NEW STACK
423  002152  022737  002122  000000            CMP     #LIMCC,@#0      ;IS RETURN ADDRESS ON NEW STACK?
424  002160  001401                            BEQ     .+4
425  002162  000000                            HLT                     ;ERROR! RETURN ADDRESS NOT SAVED ON NEW
426                                                                    ;STACK
427  002164  005077  176322                    CLR     @SLR            ;GET READY TO GET NEW VALUES FOR TEST
```

```
                                                    MOV     SPBOT,SP        ;INITIALIZE THE STACK POINTER
                                                    MOV     R4,-2(2)        ;RESTORE STACK ADDRESS DATA
                                                    MOV     R5,-4(2)
                                                    BR      T5A             ;GET NEW VALUES
002206  012767  000006  175570  LIMX:             MOV     #6,ERRVEC       ;RESTORE TIME OUT TRAP VECTOR


002214  005767  176262                           TST     ICNTA                   ;TEST FOR
002220  001013                                   BNE     END
002222  022767  177777  000120                  CMP     #-1,$ENDAD+4            ;SCRIPT
002230  001417                                   BEQ     DONE                    ;CONDITION
002232  026767  175610  175602                  CMP     46,42                   ;AND
002240  001003                                   BNE     END                     ;SET
002242  005267  176234                           INC     ICNTA                   ;FLAG
002246  000410                                   BR      DONE
002250  005267  176224          END:             INC     ICNT
002254  026727  176220  001500                  CMP     ICNT,#1500      ;HAVE 1500 PASSES BEEN COMPLETED
002262  001402                                   BEQ     DONE
002264  000167  176772                           JMP     BEGIN
002270  012767  000007  175270  DONE:            MOV     #7,TPB          ;RING BELL
002276  105767  175262                           TSTB    TPS
002302  100375                                   BPL     .-4
002304  012767  000052  175254                  MOV     #52,TPB         ;PRINT '*'
002312  105767  175246                           TSTB    TPS
002316  100375                                   BPL     .-4
002320  012767  000000  175240                  MOV     #0,TPB          ;INSERT
002326  105767  175232                           TSTB    TPS             ;NULL
002332  100375                                   BPL     .-4             ;CHARACTER
002334  013702  000042                           MOV     @#42,%2         ;GET MONITOR RETURN ADDRESS
002340  001410                                   BEQ     DONE1           ;DO NOT RETURN IF (42)=0
002342  000005                                   RESET
002344  004712          $ENDAD:                  JSR     7,(2)           ;RETURN TO MONITOR
002346  000240                                   NOP                     ;ACT11
002350  000240                                   NOP                     ;OVERLAY
002352  000240                                   NOP                     ;AREA
002354  005000                                   CLR     R0              ;DELAY FOR ACT11
002356  005200                                   INC     R0
002360  001376                                   BNE     .-2
002362  000167  176422          DONE1:           JMP     START           ;RESTART


        002366                  STKL03=.


        000001                          .END
```

```
BEGIN   001262            252     254     256     259#    270     446
DDISP = 177570            179#    206     240
DISPLA  000506            206#    240*    248*    260*
DISPRE  000174            195#    248
DONE    002270            438     442     445     447#
DONE1   002362            457     466#
DSWR  = 177570            176#    205     239
END     002250            436     440     443#
ERRVEC= 000004            184#    237     238*    249*    266*    276*    277*    291*    298*    374*    377*    392*    401*
                          410*    432*
FTITLE  000520            211#    218     234*
HLT   = 000000            180#    269     281     295     310     315     329     342     346     356     360     382     394
                          397     403     406     413     417     422     425
ICNT    000500            203#    250*    255     260     443*    444
ICNTA   000502            204#    251     257*    435     441*
LIMA    001750            377     382#
LIMAA   002010            381     384     391#
LIMB    002030            392     395#
LIMC    002064            401     404#
LIMCC   002122            413#    423
LIMD    002124            410     415#
LIMX    002206            374     432#
POP2  = 022626            181#    280     294
PRTY7 = 000340            183#    215
PSW   = 177776            174#    215*
SCOPE = 010701            182#    275     304     323     337     351     367
SLH     000514            209#    292*    331*    352*
SLR     000512            208#    278     292     306*    307     312*    313     325*    326     338*    339     344     353
                          357*    358     379*    427*
SPBOT   000510            207#    214     259     373     428
START   001010            199     214#    466
STARTA  001130            219     222     234#
STARTB  001224            217     250#
STKL01= 001364            284#    286
STKL02= 001754            386#    388
STKL03= 002366            468#    470
SWR     000504            205#    239*    241     247*    261     263
SWREG   000176            196#    247
TEMP    000516            210#    324*    327     332*    411*    415     419
TEST  = 000000            286#    388#    470#
TITLE   001064            220     227#
TORET   001330            266     271#
TPB   = 177566            178#    225*    447*    450*    453*
TPS   = 177564            177#    223     448     451     454
TO      001332            275#    282
TOA     001356            276     280#
TOB     001410            279     291#    296
TOC     001426            291     294#
TOD     001434            293     298#
T1      001444            299     304#
T1A     001452            306#    317
T2      001512            318     323#
T2A     001524            326#    333
T3      001552            337#    347
T4      001612            345     351#    361
T5      001654            359     367#
```

```
TSA      001666              370#    431
TSB      001706              374#
UBREAK=  177770              175#    263*
$ENDAD   002344              190     253     437     459#
.      = 002366              186#    187     189#    191#    194#    198#    201#    213#    224     233#    262     284     289#
                             309     314     328     341     355     386     389#    396     405     416     421     424     449
                             452     455     465     468
```

```
COMMEN          1#
ENDCOM          1#
ESCAPE          1#
GETPRI          1#
GETSWR          1#
MULT            1#
NEWTST          1#
POP             1#
PUSH            1#
REPORT          1#
SETPRI          1#
SETUP           1#
SKIP            1#
SLASH           1#
STARS           1#
SWRSU           1#      148#     235
TYPBIN          1#
TYPDEC          1#
TYPNAM          1#
TYPNUM          1#
TYPOCS          1#
TYPOCT          1#
TYPTXT          1#
$$ESCA          1#
$$NEWT          1#
$$SKIP          1#
.EQUAT          1#
.HEADE          1#
.KT11           1#
.SETUP          1#
.SWRHI          1#
.$ACT1          1#
.$APTB          1#
.$APTH          1#
.$APTY          1#
.$ASTA          1#
.$CATC          1#
.$CMTA          1#
.$DB2D          1#
.$DB2O          1#
.$DIV           1#
.$EOP           1#
.$ERRO          1#
.$ERRT          1#
.$MULT          1#
.$POWE          1#
.$RAND          1#
.$RDDE          1#
.$RDOC          1#
.$READ          1#
.$R2AZ          1#
.$SAVE          1#
.$SB2D          1#
.$SB2O          1#
.$SCOP          1#
.$SIZE          1#
```

```
.$SUPR      1#
.$TRAP      1#
.$TYPB      1#
.$TYPD      1#
.$TYPE      1#
.$TYPO      1#
.$4OCA      1#
.1170       1#


. ABS.   002366     000

 ERRORS DETECTED:  0

 CCKBFD,CCKBFD/CRF/SOL/PAGNUM/NL:TOC=CCKBFD.SML,CCKBFD.P11
 RUN-TIME:  6 6  .1 SECONDS
 RUN-TIME RATIO:  447/13=32.3
 CORE USED:  31K  (61 PAGES)
```